

On the synchronization of planar automata

J. Andres Montoya, Christian Nolasco
Universidad Nacional de Colombia, Bogota, Colombia.

December 15, 2016

Abstract

Planar automata seems to be representative of the synchronizing behavior of deterministic finite state automata. We conjecture that Černy's conjecture holds true, if and only if, it holds true for planar automata. In this paper we have gathered some evidence concerning this conjecture. This evidence amounts to show that the class of planar automata is representative of the algorithmic hardness of synchronization

This work is related to the synchronization of deterministic finite state automata (DFAs, for short).

Let \mathcal{M} be a DFA, and let $\Sigma_{\mathcal{M}}$ be its input alphabet, we use the symbol $\Sigma_{\mathcal{M}}^*$ to denote the set of finite strings over the alphabet $\Sigma_{\mathcal{M}}$. The function $\delta_{\mathcal{M}} : \Sigma_{\mathcal{M}}^* \times Q_{\mathcal{M}} \rightarrow Q_{\mathcal{M}}$ is defined by the equation:

$$\widehat{\delta_{\mathcal{M}}}(w_1 \dots w_n, q) = \delta_{\mathcal{M}}(w_n, \widehat{\delta_{\mathcal{M}}}(w_1 \dots w_{n-1}, q)),$$

where $\delta_{\mathcal{M}}$ is the transition function of \mathcal{M} .

A *synchronizing string* (reset word) for \mathcal{M} , is a string $w \in \Sigma_{\mathcal{M}}^*$ such that for all $p, q \in Q_{\mathcal{M}}$, the equality

$$\widehat{\delta_{\mathcal{M}}}(w, p) = \widehat{\delta_{\mathcal{M}}}(w, q)$$

holds

We say that automaton \mathcal{M} is *synchronizing*, if and only if, there exists a synchronizing string for \mathcal{M} . Let \mathcal{M} be a synchronizing automaton, its *minimal reset length*, denoted by $rl_{\mathcal{M}}$, is the length of its minimal synchronizing strings. It is easy to prove that $rl_{\mathcal{M}} \in O(|Q_{\mathcal{M}}|^3)$. Černy [5] conjectured that $rl_{\mathcal{M}} \leq (|Q_{\mathcal{M}}| - 1)^2$. This conjecture is called *Černy's Conjecture*, and it is considered the most important open problem in the combinatorial theory of finite state automata.

The universality conjecture for planar automata. It is well known that if Černy's conjecture holds true for strongly connected automata, then it holds true for all the deterministic finite state automata. Therefore, we say that

the class of strongly connected automata is *universal*. We conjecture that the same is true of the class of planar automata.

Let us discuss some of the facts that led us to formulate the *universality conjecture* for planar automata.

We are interested in some algorithmic problems related to DFA-synchronization. It happens that the algorithmic complexity of those problems is well understood, and there are many deep results characterizing their intrinsic hardness [6], [8], [11]. It can be noticed that all those hardness proofs work well for the planar restrictions of the problems. Then, we have that the planar restrictions of those problems are as hard as the unrestricted versions. It means that the class of planar automata is an *universal class* with respect to the algorithmic hardness of synchronization.

It can be checked that all the sequences of slowly synchronizing automata registered in the literature are sequences of planar automata (see reference [1]). At this point, it is important to remark that it is fairly easy to transform a planar sequence of slowly synchronizing automata into a nonplanar sequence with the same synchronizing behavior, and then it follows that there exist nonplanar sequences of slowly synchronizing automata. However, it seems that all the sequences of slowly synchronizing automata can be obtained this way: By locally perturbing a sequence of slowly synchronizing planar automata. Notice that if the last assertion is true, all the automata that could refute Černý's conjecture are *essentially planar*.

The above two observations are the origin of the conjecture. By the way we have found some additional evidence in favor of it. The goal of this work is to discuss in more detail those old and new facts.

Organization of the work and contributions. This work is organized into two sections, in section 1 we characterize the algorithmic hardness of some synchronization problems related to planar automata and we show that those problems are as hard as the nonplanar versions. Those results amount to show that the class of planar automata is universal with respect to the hardness of synchronization. We finish in section 2 with some concluding remarks.

1 On the algorithmic hardness of synchronizing planar automata

We investigate the synchronization of finite state automata focussing on the class of deterministic planar automata. A finite state automaton is planar, if and only if, its transition digraph is planar. Planar automata have been previously studied, and it is known that there are regular languages which cannot be recognized by deterministic planar automata [4]. This last fact indicates that the class of planar automata is not universal with respect to the recognition power of finite automata. However, we conjecture that this restricted class is universal with respect to the hardness of synchronization. This conjecture motivates us to study the synchronization of planar automata. To begin with, we

study the algorithmic complexity of some synchronization problems for planar and nonplanar automata.

Problem 1 (*Synch* [*P*]): *optimal synchronization of planar automata*)

- *Input:* (\mathcal{M}, k) , where \mathcal{M} is a synchronizing planar automaton and k is a positive integer.
- *Problem:* Decide if there exists a synchronizing string for \mathcal{M} whose length is upperbounded by k .

Theorem 2 *The problem **Synch** [*P*] is NP complete.*

Proof. Eppstein [6] proved there exists a ptime algorithm, which, on input α (where α is a CNF) computes a pair $(\mathcal{M}_\alpha, k_\alpha)$, such that \mathcal{M}_α is a synchronizing automaton satisfying the following two conditions:

1. If α is satisfiable there exists a reset word for \mathcal{M}_α , whose length is upperbounded by k_α .
2. If α is not satisfiable the length of the minimal reset words for \mathcal{M}_α is equal to $k_\alpha + 1$.

One can easily check that for all α , the automaton \mathcal{M}_α is planar. Thus, we have that set of outputs of Eppstein's reduction is included in the class of planar synchronizing automata, and it implies that SAT is ptime reducible to *Synch* [*P*]. Thus, we have that *Synch* [*P*] is NP hard. It is easy to check that *Synch* [*P*] belongs to NP ■

It was fairly easy to prove that *Synch* [*P*] is NP complete, we just noticed that all the gadgets used in Eppstein's proof are planar. A similar fact will happen more than once in this work: We get the hardness result for planar automata by noticing that the proof for general automata works verbatim in the planar framework.

How hard is the problem of approximating the minimal reset length of an automaton? First, we observe that Eppstein's greedy algorithm [6] is a ptime approximation algorithm of ratio $O(n)$, and it is clear that it must work as well when one restricts its execution to planar synchronizing automata. It is natural to ask: Which is the best approximation ratio that can be achieved in polynomial time?

Theorem 3 *Given $\varepsilon > 0$, it is NP hard to approximate the minimal reset length of planar automata within the ratio $O(n^{1-\varepsilon})$.*

Proof. Gawrychowski and Straszak [8] proved that for all $\varepsilon > 0$, it is NP hard to approximate the minimal reset length of general synchronizing automata within the ratio $O(n^{1-\varepsilon})$. Once again, it is enough to check that the proof of Gawrychowski and Straszak works for planar automata. ■

Thus, we have that the best approximation ratio that can be achieved in polynomial time is the ratio $O(n)$, which is achieved by Eppstein algorithm. Moreover, the claim is true for planar and general synchronizing automata.

It is worth to remark that the computation of minimal reset lengths is not a typical NP computation, given that minimality corresponds to an universal assertion instead of an existential assertion. Then, it cannot be said that the NP completeness of *synch* characterizes the intrinsic hardness of computing minimal reset lengths and minimal synchronizing strings. This observation motivates the study of a second algorithmic problem, denoted by *ESynch* [P] and defined by:

Problem 4 (*ESynch* [P] : Deciding minimal reset length)

- *Input:* (\mathcal{M}, k) , where \mathcal{M} is a synchronizing planar automaton and k is a positive integer.
- *Problem:* Decide if the minimal reset length of \mathcal{M} is equal to k .

Let DP be the closure under finite intersections of the class $\text{NP} \cup \text{co-NP}$, we prove that *ESynch* [P] is complete for DP. Olschewski and Ummels proved that *ESynch* [P] is complete for DP (see reference [11]). Our result is, once again, an easy consequence of the nonplanar result (with its proof), but this time we have to work a little bit.

Theorem 5 *ESynch* [P] is complete for DP.

As remarked before Olschewski and Ummels proved that *ESynch* is complete for DP. To this end, they exhibited a ptime reduction of the problem SAT-UNSAT in the problem *ESynch*. Recall that SAT-UNSAT is the problem defined by:

Proof.

- *Input:* (α, β) , where α and β are boolean formulas in conjunctive normal form.
- *Problem:* Decide if α is satisfiable and β is unsatisfiable.

It is known that the later problem is DP complete, and hence the reduction of Olschewski and Ummels suffices. Given $\alpha(X_1, \dots, X_k)$ and $\beta(X_1, \dots, X_k)$, the aforementioned reduction allows one to compute a pair $(\mathcal{M}_{\alpha\beta}, k+3)$ such that:

- If α and β are satisfiable, then the minimal reset length of $\mathcal{M}_{\alpha\beta}$ is equal to $k+2$.
- If α is satisfiable and β is unsatisfiable, then the minimal reset length of \mathcal{M} is equal to $k+3$.
- If α is unsatisfiable, then the minimal reset length of $\mathcal{M}_{\alpha\beta}$ is equal to $k+4$.

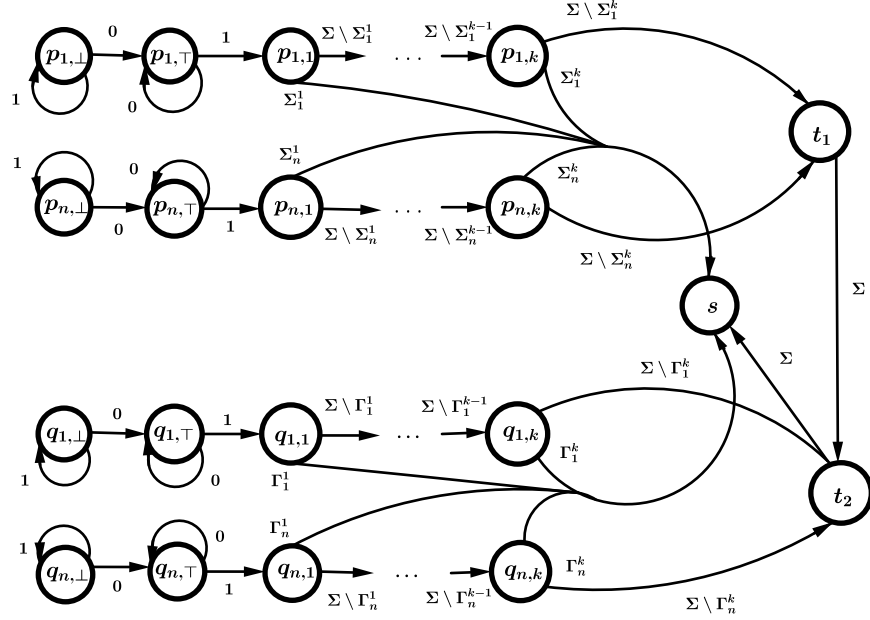


Figure 1: Olschewski-Ummels construction

The automaton $\mathcal{M}_{\alpha\beta}$ is defined as follows:

We suppose, without loss of generality, that α and β have the same number n of clauses, and no propositional variable occurs in both α and β . Let $\alpha = C_1 \wedge \dots \wedge C_n$ and $\beta = D_1 \wedge \dots \wedge D_n$. The automaton $\mathcal{M}_{\alpha\beta}$ consists of the states s, t_1, t_2, p_{ij} and q_{ij} , where $i \in \{1, \dots, n\}$ and $j \in \{\top, \perp, 1, \dots, k\}$; the transitions are depicted in figure 1. An edge from p to q labelled with $\Delta \subseteq \Sigma$ has the meaning that $\delta(p, a) = q$ for all $a \in \Delta$. The sets Σ_i^j are defined by $0 \in \Sigma_i^j \Leftrightarrow \neg X_j \in C_i$ and $1 \in \Sigma_i^j \Leftrightarrow X_j \in C_i$, and the sets Γ_i^j are defined by $0 \in \Gamma_i^j \Leftrightarrow \neg X_j \in D_i$ and $1 \in \Gamma_i^j \Leftrightarrow X_j \in D_i$.

It can be observed that $\mathcal{M}_{\alpha\beta}$ is not always a planar automaton. However, such a construction can be slightly modified in order to obtain a planar automaton $\mathcal{M}_{\alpha\beta}^*$ satisfying the above three conditions. A possible modification consists in eliminating the nodes t_1 and t_2 and replacing them with the set of nodes

$$\{p_{i,j} : i \leq n, j = k+1, k+2\} \cup \{q_{i,j} : i \leq n, j = k+1\}.$$

Moreover, for each $a \in \Sigma$ we set

$$\delta(q, a) = \begin{cases} p_{i,j+1} & \text{if } q = p_{i,j} \text{ and } j = k, k+1 \\ s, & \text{if } q = p_{i,k+2} \\ q_{i,j+1} & \text{if } q = q_{i,j} \text{ and } j = k \\ s, & \text{if } q = q_{i,k+1} \end{cases}$$

It is easy to check that the automaton so defined, which we denote with symbol $\mathcal{M}_{\alpha\beta}^*$, is a planar automaton satisfying the same three conditions satisfied by $\mathcal{M}_{\alpha\beta}$. Thus, we have a ptime reduction of SAT-UNSAT in the problem $ESynch[P]$, and hence we can conclude that the later problem is complete for DP. ■

Olschewski and Ummels also proved that the problem of computing the minimal reset length of a given synchronizing automaton is complete for the class $FNP^{\log(n)}$ [11]. It can be checked that their proof can be used (verbatim) to show that computing the minimal reset length of planar automata is complete for the same class of functions.

1.1 Synchronizing small sets of states

In this section we add a further hardness result to the above list. This time we have to work hard, because we have to prove first the corresponding nonplanar result, which characterizes the parameterized complexity of *subset synchronization*. We refer the reader to [7] for a pedagogical introduction to the basics of parameterized complexity.

Given an automaton \mathcal{M} , and given $q_1, \dots, q_k \in Q_{\mathcal{M}}$, a synchronizing string for those k states, is a string w , such that for all $i, j \leq k$, the equality

$$\widehat{\delta_{\mathcal{M}}}(w, q_i) = \widehat{\delta_{\mathcal{M}}}(w, q_j)$$

holds. In the later case we say that w synchronizes the subset $\{q_1, \dots, q_k\}$.

We think that subset synchronization is a powerful concept that allows one to model some interesting discrete dynamics. Suppose, for instance, that one has a troop of agents scattered over a territory, and that he wants to broadcast an instruction, the same one for all the agents, which must lead the agents to a common site on the territory. If the territory is the transition digraph of a synchronizing automaton, and he does not know the initial locations of the agents, then he must broadcast a reset word for the underlying automaton. On the other hand, if he knows the initial locations q_1, \dots, q_k , then he must broadcast a synchronizing string for these k states.

A second dynamics refers a class of games on checkerboards. Suppose one has a set of tokens scattered over checkerboard. Each time he chooses an available action, it determines the way in which he must move each one of his tokens. Tokens that arrive to the same site are stacked. The goal is to gather all the tokens into a single stack using as few actions as possible. Notice that if the transition digraph defined by the checkerboard together with the set of allowed actions defines the transition digraph of an automaton, say \mathcal{M} , then the goal

of this game corresponds to synchronize the states of \mathcal{M} that were occupied by tokens at time 0.

Notice that, in the above two situations, one is interested in computing minimal synchronizing strings for the set of occupied states. We consider that planar automata are natural scenarios for the most representative instances of those two problems: On one hand, it can be argued that planar digraphs, which are the discrete versions of the surfaces of genus 0, are natural discrete models of territories. And, on the other hand, most checkerboards are planar.

First, we consider the case where the number of states to be synchronized is upperbounded by a fixed constant. We use the symbol $k\text{-Synch}$ to denote the problem.

Problem 6 ($k\text{-synch}$: Synchronization of k -tuples)

- *Input:* $(\mathcal{M}, \{q_1, \dots, q_r\})$, where \mathcal{M} is a synchronizing automaton, $q_1, \dots, q_r \in Q_{\mathcal{M}}$ and $r \leq k$.
- *Problem:* Compute a synchronizing string of minimal length for the states q_1, \dots, q_r .

Given $k \geq 2$, it is easy to show that the problem $k\text{-Synch}$ can be solved in polynomial time.

Theorem 7 $k\text{-Synch}$ can be solved in time $O(|\mathcal{M}|^{2k})$.

Proof. Let $(\mathcal{M}, \{q_1, \dots, q_r\})$ be an instance of $k\text{-Synch}$, consider the k -power automaton \mathcal{M}^k defined by:

- $Q_{\mathcal{M}^k} = \{\{p_1, \dots, p_k\} : p_1, \dots, p_k \in Q_{\mathcal{M}}\}$ (the p 's are not necessarily pairwise different).
- Given $c \in \Sigma_{\mathcal{M}}$, the equality

$$\delta_{\mathcal{M}^k}(c, \{p_1, \dots, p_k\}) = \{\delta_{\mathcal{M}}(c, p_1), \dots, \delta_{\mathcal{M}}(c, p_k)\}$$

holds.

Computing a minimal synchronizing string for states q_1, \dots, q_r , is the same as computing a minimal path in \mathcal{M}^k , connecting the state $\{q_1, \dots, q_r\}$ with the set $\Delta_{\mathcal{M}^k} = \{A \in Q_{\mathcal{M}^k} : |A| = 1\}$. The later problem can be solved in time $O(|\mathcal{M}|^{2k})$. ■

Notice that, when estimating the running time of the above algorithm, the parameter k occurs in the exponent. What does it happens if the parameter k is not fixed? Let $p\text{-Synch}[P]$ be the parameterized problem defined by:

Problem 8 ($p\text{-synch}[P]$: Parameterized synchronization of planar automata)

- *Input:* $(\mathcal{M}, \{q_1, \dots, q_k\}, k, r)$, where \mathcal{M} is a synchronizing planar automaton and $q_1, \dots, q_k \in Q_{\mathcal{M}}$.

- *Parameter:* k .
- *Problem:* Decide if there exists a synchronizing string of length r for the states q_1, \dots, q_k .

Recall that a parameterized problem is fix parameter tractable, if and only if, it can be solved in time $O(f(k) \cdot n^c)$, for some function f and some constant c (see reference [7]). Is $p\text{-Synch}[P]$ fix parameter tractable? We prove that the problem $p\text{-Synch}[P]$ is NWL complete. The class NWL is supposed to be the parameterized analogue of PSPACE [10]. This class is located above of the W-hierarchy, and hence our result implies that $p\text{-Synch}[P]$ is $W[t]$ hard for all $t \geq 1$. The class NWL is defined as the closure under *fpt-reductions* (see reference [7]) of the following problem.

Problem 9 (*p-NWL : deciding acceptance of parameterized space bounded computations*)

- *Input:* (\mathcal{M}, t, k) , where \mathcal{M} is a nondeterministic Turing machine, t is a positive integer given in unary, and $k \geq 1$.
- *Parameter:* k .
- *Problem:* Decides if \mathcal{M} accepts the empty input in at most t steps and checking at most k cells.

Given a parameterized problem L , if one wants to check that L belongs to NWL , it is enough to exhibit a nondeterministic RAM accepting the language L , and such that the number of registers it uses along the computation, on input X , is bounded above by a quantity that only depends on the parameter of X (see [10]). We prove that $p\text{-Synch}[P]$ is NWL hard by exhibiting an fpt Turing reduction of *The parameterized longest common subsequence problem* in $p\text{-Synch}[P]$. The parameterized longest common subsequence problem, denoted by $p\text{-LCS}$, is the parameterized problem defined by:

Problem 10 (*p-LCS : parameterized longest common subsequence*)

- *Input:* $(\{w_1, \dots, w_k\}, \Sigma, m)$, where Σ is a finite alphabet, $w_1, \dots, w_k \in \Sigma^*$ and m is a positive integer.
- *Parameter:* k
- *Problem:* Decide if there exists a string $w \in \Sigma^*$, such that for all $i \leq k$ string w is a substring of w_i , and such that $|w| = m$.

Guillemot [10] proved that $p\text{-LCS}$ is hard for NWL .

Theorem 11 *The problems $p\text{-Synch}[P]$ and $p\text{-Synch}$ are NWL complete.*

Proof. First, we check that $p\text{-Synch}$ belongs to NWL . To this end, we construct a suitable nondeterministic RAM accepting the problem $p\text{-Synch}$. The machine works, on input, $(\mathcal{M}, \{q_1, \dots, q_k\}, l, k)$, as follows:

The machine stores in the first k registers a tuple of positive integers (s_1, \dots, s_k) , such that for all $i \leq k$ the inequality $s_i \leq |Q|$ holds. It begins with $(0, \dots, 0)$, and then it overwrites (q_1, \dots, q_k) . Set $(s_1^1, \dots, s_k^1) = (q_1, \dots, q_k)$, for all $i \leq l$ the machine nondeterministically chooses a tuple $(s_1^{i+1}, \dots, s_k^{i+1})$, which can (over)writes on the first k registers, if and only if, there exists $a \in \Sigma$ such that $\delta(a, s_j^i) = s_j^{i+1}$. The machine accepts if and only if the entries of the last tuple are all equal.

Now, we prove the NWL hardness of $p\text{-Synch}[P]$ and $p\text{-Synch}$. First, we prove that $p\text{-LCS}$ is fpt many-one reducible to $p\text{-Synch}$, and hence we prove that $p\text{-Synch}$ is fpt Turing reducible to $p\text{-Synch}[P]$. The later reduction is given as the composition of two reductions. The first one is a fpt many-one reduction of the problem $p\text{-Synch}$ in the problem $p\text{-Synch}[2]$, which is the restriction of $p\text{-Synch}$ to binary automata (automata whose input alphabet has size 2). The second one is a fpt many-one reduction of $p\text{-Synch}[2]$ in $p\text{-Synch}[P]$.

First stage (*Reducing $p\text{-LCS}$ to $p\text{-Synch}$*).

Let $X = (\{w_1, \dots, w_k\}, \Sigma, m)$ be an instance of $p\text{-LCS}$. Given $i \leq k$, we use Baeza-Yates construction (see [2]) to compute a DFA, say \mathcal{M}_i , that accepts the language constituted by all the subsequences of w_i . It is important to remark that the size of \mathcal{M}_i is bounded above by $|w_i| + 1$.

Notice that for all $i \leq k$, we are using the automaton \mathcal{M}_i as a language acceptor, it implies that for all $i \leq k$, there exists a marked state (the initial state of \mathcal{M}_i) which we denote with the symbol q_0^i . Moreover, for all $i \leq k$, there exists a nonempty subset of Q_i , denoted with the symbol A_i , and which is equal to the set of accepting states of automaton \mathcal{M}_i .

We use the set $\{\mathcal{M}_i : i \leq k\}$ to define an automaton $\mathcal{M} = (\Omega, Q, \delta)$ in the following way:

1. $\Omega = \Sigma \cup \{d\}$, where $d \notin \Sigma$.
2. $Q = \left(\bigsqcup_{i \leq k} Q_i \right) \sqcup \{q, p_1, \dots, p_{m+1}\}$, where \sqcup denotes disjoint union, and given $i \leq k$, the symbol Q_i denotes the set of states of the automaton \mathcal{M}_i . Moreover, we have that $q, p_1, \dots, p_{m+1} \notin \bigsqcup_{i \leq k} Q_i$.
3. The transition function of \mathcal{M} , which we denote with the symbol δ , is

defined as follows

$$\delta(p, a) = \begin{cases} \delta_i(p, a), & \text{if } p \in Q_i \text{ and } a \neq d \\ q, & \text{if } p \in \bigsqcup_{i \leq k} A_i \text{ and } a = d \\ p_1, & \text{if } p \in (Q_i \setminus A_i) \text{ and } a = d \\ q, & \text{if } p = q \\ p_{j+1}, & \text{if } p = p_j, j < m+1 \text{ and } a \in \Sigma \\ p_1, & \text{if } p = p_j, j < m+1, \text{ and } a = d \\ q, & \text{if } p = p_{m+1} \text{ and } a = d \\ p_1, & \text{if } p = p_{m+1} \text{ and } a \neq d \end{cases}$$

Let $Y(X)$ be equal to $(\mathcal{M}, \{q_0^1, \dots, q_0^k, p_1\}, k+1, m+1)$, we have that $Y(X)$ it is the output of the first reduction. It is easy to check that $X \in p\text{-LCS}$, if and only if, the states q_0^1, \dots, q_0^k, p_1 can be synchronized in time $m+1$, that is: It can be easily checked that $X \in p\text{-LCS}$, if and only if, $Y(X) \in p\text{-Synch}$.

Unfortunately, it happens that Baeza-Yates construction is nonplanar, and hence if $Y(X)$ is equal to $(\mathcal{M}, \{q_0^1, \dots, q_0^k, p_1\}, k+1, m+1)$, it could occur that the automaton \mathcal{M} is a nonplanar one. Therefore, we have to proceed with the second reduction.

Second stage (*Reducing $p\text{-Synch}$ to $p\text{-Synch}[P]$*)

Let $p\text{-Synch}[2]$ be the restriction of $p\text{-Synch}$ to the set of instances

$$\{(\mathcal{M}, \{q_1, \dots, q_k\}, l, k) : \mathcal{M} \text{ is binary synchronizing automaton}\}.$$

The construction used in [3] yields a *fpt* many-one reduction of the problem $p\text{-Synch}$ in its restriction $p\text{-Synch}[2]$. Now, we will exhibit a *fpt* Turing reduction of the problem $p\text{-Synch}[2]$ in the problem $p\text{-Synch}[P]$.

Let $(\mathcal{M}, \{q_1, \dots, q_k\}, m)$ be an instance of $p\text{-Synch}[2]$. A planar drawing of the automaton \mathcal{M} is an embedding in \mathbb{R}^2 of its transition digraph, and which satisfies the following three properties:

- Edges are mapped on simple curves.
- No three edges meet at a common crossing.
- Two edges meet at most once.

Planar drawings can be computed in polynomial time, and if the automaton \mathcal{M} is a planar one hence its planar drawing can be chosen to be a planar embedding. Suppose that \mathcal{M} is nonplanar, and let ρ be a planar drawing of \mathcal{M} . Given e an edge (transition) of \mathcal{M} , we use the symbol $cr_\rho(e)$ to denote the number of crossings involving edge e . Notice that for all ρ and for all e the inequality $cr_\rho(e) \leq 2|Q|$ holds.

To begin with the reduction we compute a planar drawing of \mathcal{M} , say ρ , and we use ρ to compute a planar automaton \mathcal{N} . The computation of \mathcal{N} goes as follows:

1. Let $\{a, b\}$ be the input alphabet of \mathcal{M} , the input alphabet of \mathcal{N} is equal to $\{a, b\} \times \{0, 1\}$.
2. Let e be an edge of \mathcal{M} , and suppose that e is labeled with the letter a . We partition $\rho(e)$ into $2|Q|$ disjoint segments. The idea is to build $2|Q|$ gadgets that are used to eliminate the crossings involving e . The segments can be chosen to be connected, with a nonempty interior, and such that each crossing is an inner point of one of those intervals. Moreover, we can choose the $2|Q|$ segments in such a way that each one of them contains at most one crossing. The gadgets are extremely simple:

Suppose that e is directed from p to q . We observe that each one of the $2|Q|$ segments has a first point (the closest to $\rho(p)$). Given $1 \leq i \leq 2|Q|$, we choose four points in the i th segment. Let $v_1^{e,i}, v_2^{e,i}, v_3^{e,i}$ and $v_4^{e,i}$ be those four points, we have that $v_1^{e,i}$ is equal to the first point of the segment, point $v_2^{e,i}$ lies between $v_1^{e,i}$ and $v_3^{e,i}$, while $v_3^{e,i}$ lies between $v_2^{e,i}$ and $v_4^{e,i}$. Moreover, the point $v_4^{e,i}$ is different to the first point of the $(i+1)$ th segment. If $i = 1$, we have that $v_1^{e,i} = \rho(p)$. If $i = 2|Q|$, we set $v_1^{e,i+1} = \rho(q)$. Let $i \leq 2|Q|$, notice that the i th segment has been divided into four subsegments e_1^i, e_2^i, e_3^i and e_4^i . Given $j \leq 3$, the edge e_j^i is directed from $v_j^{e,i}$ to $v_{j+1}^{e,i}$, while the edge e_4^i is directed from $v_4^{e,i}$ to $v_1^{e,i+1}$. Moreover, we assign to those four edges the labels $(a, 0), (a, 1), (a, 1)$ and $(a, 0)$ (respectively).

3. Now suppose that edges e and f meet at some point x . There exists $i, j \leq 2|Q|$ such that x lies on the i th segment of e , and it also lies on the j th segment of f . We can choose the points $v_1^{e,i}, v_2^{e,i}, v_3^{e,i}$ and $v_4^{e,i}$, and the points $v_1^{f,j}, v_2^{f,j}, v_3^{f,j}$ and $v_4^{f,j}$ in such a way that:

- The equalities $v_3^{e,i} = v_2^{f,j}$ and $v_4^{e,i} = v_3^{f,j}$ hold.
- The point x lies between $v_3^{e,i}$ and $v_4^{e,i}$

Notice that the construction is somewhat asymmetrical. However, it does not matter: Given two edges that cross each other at some point x , it makes not difference which edge plays the role of edge e and which one plays the role of edge f .

Suppose that f is labeled with the letter b , and suppose that $b \neq a$, in this case the elimination of the crossing looks as follows

If $b = a$, we merge the two parallel edges going from $v_3^{e,i}$ ($v_2^{f,j}$) to $v_4^{e,i}$ ($v_3^{f,j}$), and which are labeled with the letter $(a, 1)$.

4. We are trying to draw a planar automaton \mathcal{N} . To this end, we use the set of points

$$P = \left\{ v_j^{e,i} : i \leq 2|Q|, j = 1, 2, 3, 4 \text{ and } e \text{ is an edge of } \mathcal{M} \right\},$$

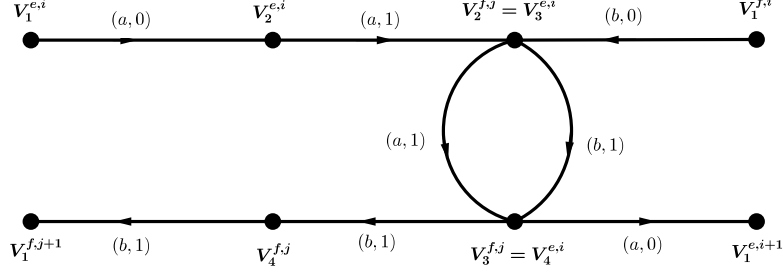


Figure 2: Elimination of the crossing

and the set of edges

$$E = \{e_j^i : i \leq 2|Q|, j = 1, 2, 3, 4 \text{ and } e \text{ is an edge of } \mathcal{M}\},$$

and we add some loops: If $v \in P$, and there is not an outgoing edge labeled with the letter $x \in \Sigma \times \{0, 1\}$, we add a loop with this label attached to this node.

It is easy to check that \mathcal{N} is a planar synchronizing automaton, which can be computed in polynomial time from automaton \mathcal{M} . Notice that the construction of \mathcal{N} does not depend on the pair $(\{q_1, \dots, q_k\}, m)$.

Let $m \geq 1$ and let p be a state of \mathcal{M} , we use the symbol $\mathcal{N}_{m,p}$ to denote the planar automaton that is obtained from \mathcal{N} by attaching to node $\rho(p)$ a planar digraph that we call \mathcal{C}_m . The gadget \mathcal{C}_m is computed from m and $|Q|$, and it is used as a clock for the synchronization process. The construction of \mathcal{C}_m goes as follows:

We begin with a set W_m , which is equal to

$$\{w_{j,\epsilon}^{i,k} : i \leq 2|Q|, k \leq m, j = 1, 2, 3, 4, \text{ and } \epsilon = 0, 1\} \cup \{w(p), \phi(p)\}.$$

Then, for all $k \leq m$ we identify the points $w_{j,0}^{1,k}$ and $w_{j,1}^{1,k}$. Given $i \leq 2|Q|$ and

given $\epsilon \in \{a, b\}$, we add the edges $(w_{1,\epsilon}^{i,k}, w_{2,\epsilon}^{i,k})$ and $(w_{4,\epsilon}^{i,k}, w_{1,\epsilon}^{i+1,k})$ and we label them with the letter $(\epsilon, 0)$. We also add the edges $(w_{2,\epsilon}^{i,k}, w_{3,\epsilon}^{i,k})$ and $(w_{3,\epsilon}^{i,k}, w_{4,\epsilon}^{i,k})$, and we label them with the letter $(\epsilon, 1)$. We set $w_{1,0}^{1,m+1} = w_{1,1}^{1,m+1} = \phi(p)$. If $i = 2|Q|$, we add the edges $(w_{4,\epsilon}^{i,k}, w_{1,\epsilon}^{1,k+1})$ and we label them with the letter $(\epsilon, 0)$. Moreover, for all $\epsilon \in \{a, b\}$ we add an edge $(w(p), w_{1,\epsilon}^{1,1})$ labeled with the letter $(\epsilon, 0)$. Finally, we add the necessary loops in order to get a planar automaton, this planar automaton (its transition digraph) is the clock \mathcal{C}_m .

Given the node $\rho(p)$, we embed \mathcal{C}_m in the plane in such a way that the following two conditions are satisfied:

1. $\phi(p) = \rho(p)$.
2. There are not crossings between the edges of \mathcal{C}_m and the edges of \mathcal{N} .

Then, we remove the loops that were attached to $\phi(p)$ in order to get a deterministic planar synchronizing automaton that we denote with the symbol $\mathcal{N}_{m,p}$.

We observe that if a token is placed on state $w(p)$, then one can move this token to the state $\rho(p)$ by using a string of length $8m|Q|$. The quantity $8m|Q|$ is the length of the shortest strings satisfying the equality

$$\widehat{\delta_{\mathcal{N}_{m,p}}}(w(p), X) = \rho(p).$$

Moreover, if the string $X \in (\{a, b\} \times \{0, 1\})^{8m|Q|}$ satisfies the above equality, there exist X_1, \dots, X_m such that $X = X_1 \cdots X_m$ and for all $i \leq m$ the factor X_i satisfies the equality

$$X_i = ((\epsilon_i, 0)(\epsilon_i, 1)(\epsilon_i, 1)(\epsilon_i, 0))^{2|Q|},$$

for some $\epsilon_i \in \{a, b\}$. That is, given $f : \{0, 1\}^* \rightarrow ((\{0, 1\} \times \{0, 1\}))^*$, the homomorphism defined by

$$f(a_1 \dots a_k) = ((a_1, 0)(a_1, 1)(a_1, 1)(a_1, 0))^{2|Q|} \dots ((a_k, 0)(a_k, 1)(a_k, 1)(a_k, 0))^{2|Q|},$$

it happens that X is a minimal string satisfying the equality

$$\widehat{\delta_{\mathcal{N}_{m,p}}}(w(p), X) = \rho(p),$$

if and only if, there exists $W_X \in \{0, 1\}^m$ such that $X = f(W_X)$. This property of \mathcal{C}_m allows us to use it as a clock: If one wants to synchronize the states $p_1, \dots, p_k, w(p)$ in less than $8m|Q| + 1$ steps, then he must try to move all those states to $\rho(p)$, and to this end he has to use a string in the range of the homomorphism f . We use this fact to avoid that some token being synchronized uses the crossing-gadgets to leave an edge of \mathcal{M} that it has not fully traversed, recall that the edges of \mathcal{M} were partitioned in many different segments (subedges).

Given the automaton \mathcal{M} and given $S = \{q_1, \dots, q_k\}$, we use the symbol $\mathcal{I}_{p,m,S}$ to denote the tuple

$$(\mathcal{N}_{m,p}, \{\rho(q_1), \dots, \rho(q_k), w(p)\}, 8m|Q|).$$

It is easy to check that the set S can be synchronized in time m , if and only if, there exists p such that the states $\rho(q_1), \dots, \rho(q_k), w(p)$ of the automaton $\mathcal{N}_{m,p}$ can be synchronized in time $8m|Q|$. Thus, we have the claimed *fpt* Turing reduction of $p\text{-Synch}$ [2] in $p\text{-Synch}[P]$, and hence we have that $p\text{-Synch}[P]$ is *NWL* hard.

It is worth to remark that the above reduction shows that $p\text{-Synch}[P, 4]$ is *NWL* hard. We use the symbol $p\text{-Synch}[P, 4]$ to denote the restriction of $p\text{-Synch}[P]$ to the class of automata defined over a four letter alphabet. ■

Given $L[P]$, one of the algorithmic problems studied so far, we use the symbol L to denote its unrestricted (nonplanar) version, i.e. symbol L denotes the algorithmic problem that is obtained from $L[P]$ by flipping the planarity constraint. Let us summarize all the above results with the following table

$\text{Synch}[P]$	$\text{Esynch}[P]$	Approx. ratio of $\text{Synch}[P]$	$p\text{-Synch}[P]$
NP complete	DP complete	$O(n)$	<i>NWL</i> complete
Synch	Esynch	Approx. ratio of Synch	$p\text{-Synch}$
NP complete	DP complete	$O(n)$	<i>NWL</i> complete

The above table seems to indicate that the class of planar automata is universal with respect to the algorithmic complexity of synchronization. Perhaps, the only issue that remains to be analyzed is the parameterized approximability of subset synchronization.

We say that $p\text{-Synch}$ is *fpt approximable within the ratio* $f(n, k)$, if and only if, there exists an *fpt* algorithm, which, on input $(\mathcal{M}, \{q_1, \dots, q_k\}, k)$, outputs an integer t such that if $rl_{\mathcal{M}}(q_1, \dots, q_k)$ is the minimal reset length of the states q_1, \dots, q_k , then the inequalities

$$rst_{\mathcal{M}}(q_1, \dots, q_k) \leq t \leq f(n, k) \cdot rst_{\mathcal{M}}(q_1, \dots, q_k)$$

hold.

It follows from the work of Gerbush and Heeringa that $p\text{-Synch}$ is *fpt* approximable within the ratio $\left\lceil \frac{n-1}{k-1} \right\rceil$ (see [9]). It is natural to ask: Which are the approximation ratios that can achieved in *fpt* time? Which are the approximation ratios that can achieved in *fpt* time for planar automata? We think that those two questions are the questions that remain to be solved, and that are related to the algorithmic complexity of synchronizing planar and nonplanar automata.

We observe that subset synchronization makes sense for nonsynchronizing automata. It is easy to check that for all $k \geq 2$ the synchronizing times of the hardest k -tuples of states is $\Omega(n^k)$ (see [12]). Synchronizing times of order $\Omega(n^k)$ are achieved by sequences of nonsynchronizing automata (an upper bound $O(k \cdot n^2)$ holds for synchronizing automata). It is also easy to check that for all $k \geq 2$ there exist sequences of planar automata achieving those worst synchronizing times of order $\Omega(n^k)$. Thus, we have that the slowest nonsynchronizing automata are planar. We conjecture that an analogous fact holds for the synchronization of all the states (whenever it is possible): The slowest synchronizing automata are planar automata.

2 Concluding remarks: Synchronizing times and The Černý Conjecture for planar automata

The hardness of a class of synchronizing automata can be measured in many different ways, we propose two different hardness measures:

- The computational hardness of the algorithmic problems (restrictions) that are determined by the class.
- The synchronizing times required by the automata within the class.

According to the first measure, the class of planar synchronizing automata is as hard as the class constituted by all the synchronizing automata. We conjecture that the same is true for the second measure.

The *weak Černý conjecture* is the conjecture claiming that there exists a quadratic polynomial $q(X)$ such that the synchronizing time of any synchronizing automata with n states is upperbounded by $q(n)$. Černý's conjecture claims that $q(n)$ can be taken equal to $(n-1)^2$.

Conjecture 12 *Given $\varepsilon > 0$, if there exists a sequence of synchronizing automata whose synchronizing time is $\Omega(n^{2+\varepsilon})$, then there must exist a sequence of planar synchronizing automata whose synchronizing time is $\Omega(n^{2+\varepsilon})$.*

We notice that our conjecture implies that The weak Černý conjecture is true, if and only if, it holds true for planar synchronizing automata. In order to prove the above conjecture one can try a construction similar to the used in the proof of theorem 11.

Let \mathcal{M} be a given nonplanar automaton, and let \mathcal{N} be the output of the aforementioned construction. We have that $rl_{\mathcal{N}} \geq 8n \cdot rl_{\mathcal{M}}$, where n is the size of \mathcal{M} . Thus, the reset length of \mathcal{N} is large provided that the reset length of \mathcal{M} is large. The problem is that the size of \mathcal{N} is quadratic with respect to the size of \mathcal{M} . Our construction does not work because of this quadratic blow-up. It could work if we could restrict its application to sequences of automata of bounded genus. In this later case we would have to use smaller clocks, clocks whose sizes are linearly related to the sizes of the automata given as input.

We notice that from a naive point of view our conjecture must be true: Planarity is a constraint that makes harder the movement of tokens through the digraph. This naive observation was an additional motivation for our conjecture, which asserts that the worst synchronizing times are achieved by planar automata. We have some further reasons to consider that it is a likely conjecture:

- As remarked before, all the sequences of slowly synchronizing automata registered in the literature are sequences of planar automata. Thus, the slowest synchronizing automata registered in the literature are planar automata.
- As it was remarked at the end of last section, the slowest nonsynchronizing automata are planar.

We would like to finish this work by proposing two problems:

- **Problem 1:** Prove conjecture 12.
- **Problem 2:** Prove The weak Černý Conjecture for planar automata.

It should be clear that positive solutions to both problems entail a proof of The weak Černý Conjecture for general synchronizing automata.

Acknowledgement. The second author would like to thank the support provided by Universidad Nacional de Colombia through the project Hermes 8943 (32083).

References

- [1] D. Ananichev, V. Gusev, M. Volkov. Slowly Synchronizing Automata and Digraphs. MFCS 2010, pages 55-65.
- [2] R. Baeza-Yates. Searching Subsequences. Theor. Comput. Sci. 78(2): 363-376, 1991.
- [3] M. Berlinkov. On Two Algorithmic Problems about Synchronizing Automata. DLT 2014, pages 61-67.
- [4] R. Book, A. Chandra. Inherently Nonplanar Automata. Acta Inf. 6: 89-94, 1976.
- [5] J. Černý. Poznámka k. homogénnym experimentom s konečnými automatmi, Mat. fyz. čas SAV 14: 208–215, 1964.
- [6] D. Eppstein. Reset sequences for monotonic automata, SIAM J. Comput. 19: 500–510, 1990.
- [7] J. Flum, M. Grohe. *Parameterized Complexity Theory*. Springer Verlag, Berlin, 2006.

- [8] P. Gawrychowski, D. Straszak. Strong Inapproximability of the Shortest Reset Word. MFCS 2015, pages 243-255.
- [9] M. Gerbush, B. Heeringa. Approximating Minimum Reset Sequences. CIAA 2010, pages 154-162.
- [10] S. Guillemot. Parameterized complexity and approximability of the Longest Compatible Sequence problem. Discrete Optimization 8(1): 50-60, 2011.
- [11] J. Olschewski, M. Ummels. The Complexity of Finding Reset Words in Finite Automata. MFCS 2010, pages 568-579.
- [12] V. Vorel. Subset Synchronization of Transitive Automata. AFL 2014, pages 370-381.